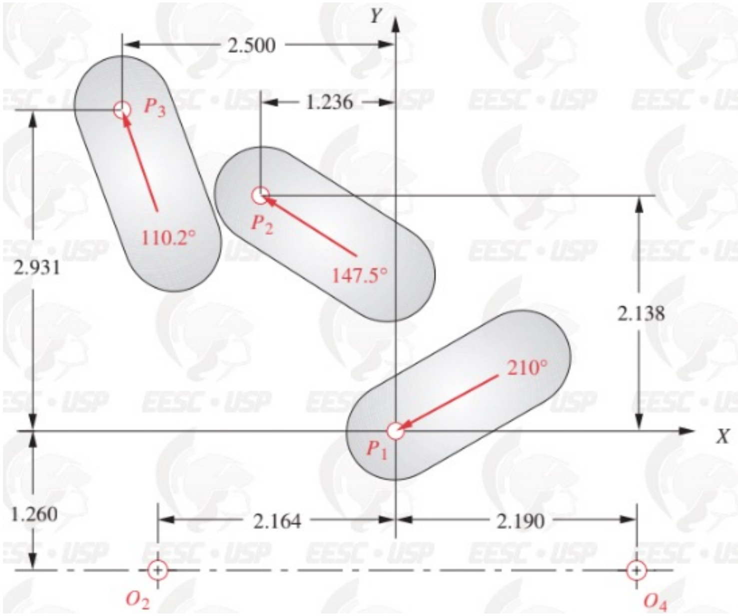
Prática 5 - Thallys Oliveira - 11819827

O objetivo desta prática é a síntese de um mecanismo de 4 barras, o qual posicione o elo de acoplamento nas posições P_1, P_2 e P_3, de acordo com a imagem ao lado. Os dados a seguir são utilizados para a análise do mecanismo.



- $P_{1x}=0$, $P_{1y}=0$; • $P_{2x}=1.236$, $P_{2y}=2.138$
- $P_{3x}=2.500$; $P_{3y}=2.931$
- $\beta_2=30°$, $\beta_3=60$; • $\gamma_2=-10°$, $\gamma_3=25°$
- $\theta_{P1}=210°$, $\theta_{P2}=147,5$ e $\theta_{P3}=110,2°$

Continuando, calculamos a magnitude dos vetores pn, como também os ângulos α e θ. Utilizamos as equações abaixo:

- $P_{mn}=\sqrt{(P_{mx})^2+(P_{my})^2}$, $\alpha_n=\theta_{pn}-\theta_{p1}$ e $\delta_m=\tan^{-1}\left(\frac{P_{mx}}{P_{my}}\right)+\frac{\pi}{2}$

Calculamos os comprimentos dos elos, dos ângulos dos elos e demais ângulos. Continuando, calculamos as coordenadas de orientação dos pivôs. Então, calculamos o máximo e mínimo de θ_2 e, via o método de Newton-Raplison, calculamos os θ_3 e θ_4.

$$\begin{bmatrix} -b\cdot\sin\theta_3 & c\cdot\sin\theta_4 \\ b\cdot\cos\theta_3 & -c\cdot\cos\theta_4 \end{bmatrix}\cdot\begin{bmatrix} \Delta\theta_3 \\ \Delta\theta_4 \end{bmatrix}=\begin{bmatrix} a\cdot\cos\theta_4+b\cdot\cos\theta_3-c\cdot\cos\theta_4-d \\ a\cdot\sin\theta_2+b\cdot\sin\theta_3-c\cdot\sin\theta_4 \end{bmatrix}$$

$$A=\cos\beta_2-1; \quad B=\sin\beta_2; \quad C=\cos\alpha_2-1$$
$$D=\sin\alpha_2; \quad E=p_{21}\cos\delta_2; \quad F=\cos\beta_3-1$$
$$G=\sin\beta_3; \quad H=\cos\alpha_3-1; \quad K=\sin\alpha_3$$
$$L=p_{31}\cos\delta_3; \quad M=p_{21}\sin\delta_2; \quad N=p_{31}\sin\delta_3$$

$$A=\cos\gamma_2-1; \quad B=\sin\gamma_2; \quad C=\cos\alpha_2-1$$
$$D=\sin\alpha_2; \quad E=p_{21}\cos\delta_2; \quad F=\cos\gamma_3-1$$
$$G=\sin\gamma_3; \quad H=\cos\alpha_3-1; \quad K=\sin\alpha_3$$
$$L=p_{31}\cos\delta_3; \quad M=p_{21}\sin\delta_2; \quad N=p_{31}\sin\delta_3$$

$$AW_{1_x}-BW_{1_y}+CZ_{1_x}-DZ_{1_y}=E$$
$$FW_{1_x}-GW_{1_y}+HZ_{1_x}-KZ_{1_y}=L$$
$$BW_{1_x}+AW_{1_y}+DZ_{1_x}+CZ_{1_y}=M$$
$$GW_{1_x}+FW_{1_y}+KZ_{1_x}+HZ_{1_y}=N$$

$$AU_{1_x}-BU_{1_y}+CS_{1_x}-DS_{1_y}=E$$
$$FU_{1_x}-GU_{1_y}+HS_{1_x}-KS_{1_y}=L$$
$$BU_{1_x}+AU_{1_y}+DS_{1_x}+CS_{1_y}=M$$
$$GU_{1_x}+FU_{1_y}+KS_{1_x}+HS_{1_y}=N$$

```
% Prática 5 - Thallys Oliveira - 11819827

% Dados dos problema

P1_x = 0; P2_x = -1.236; P3_x = -2.500;
P1_y = 0; P2_y = +2.138; P3_y = +2.931;

beta_2 = 30; beta_3 = 60;
gama_2 = -10; gama_3 = 25;
theta_P1 = 210; theta_P2 = 147.5; theta_P3 = 110.2;

% Criando vetores com os dados

P1 = [P1_x P1_y]; P2 = [P2_x P2_y]; P3 = [P3_x P2_y];
beta = [0 beta_2 beta_3];
gama = [0 gama_2 gama_3];
theta = [theta_P1 theta_P2 theta_P3];

% Transformando os ângulos de grau para radianos

beta = (pi/180)*beta;
gama = (pi/180)*gama;
theta = (pi/180)*theta;

% Cálculo iniciais

P_21 = sqrt(P2(1)^2+ P2(2)^2);
P_31 = sqrt(P3(1)^2+ P3(2)^2);

delta = [0 (-1)*atan2(P2(1),P2(2))+pi/2 (-1)*atan2(P3(1),P3(2))+pi/2]; % delta = [ Δ1↙
Δ2 Δ3]
alpha = [0 theta(2)- theta(1) theta(3)-theta(1)]; % alpha = [α1 α2 α3]

% Constantes do Sistema 1

A = cos(beta(2)) -1; B = sin(beta(2));
C = cos(alpha(2))-1; D = sin(alpha(2));
E = P_21*cos(delta(2));
F = cos(beta(3)) - 1; G = sin(beta(3));
H = cos(alpha(3)) -1; K = sin(alpha(3));
L = P_31*cos(delta(3));
M = P_21*sin(delta(3)); N = P_21*sin(delta(3));

% Sistema 1

M1 = [[A -B C -D];
      [F -G H -K];
      [B A D C];
      [G F K H];];

v = [E L M N]';

x1 = M1\v;

W1 = [x1(1) x1(2)];
```

```
Z1 = [x1(3) x1(4)];

% Magnitude e Orientação de W e Z

W_mag = sqrt(W1(1)^2+W1(2)^2);
Z_mag = sqrt(Z1(1)^2+Z1(2)^2);

t   = (-1)*atan2(W1(1),W1(2))+pi/2;
phi = (-1)*atan2(Z1(1),Z1(2))+pi/2;

% Constantes do Sistema 2

A2 = cos(gama(2)) - 1; B2 = sin(gama(2));
F2 = cos(gama(3)) - 1; G2 = sin(gama(3));

% Sistema 1

M2 = [[A2 -B2 C -D];
      [F2 -G2 H -K];
      [B2 A2 D C];
      [G2 F2 K H];];

x2 = M2\v;

U1 = [x2(1) x2(2)];
S1 = [x2(3) x2(4)];

% Magnitude e Orientação de U e S

U_mag = sqrt(U1(1)^2+U1(2)^2);
S_mag = sqrt(S1(1)^2+S1(2)^2);

sigma = (-1)*atan2(U1(1),U1(2))+pi/2;
psi   = (-1)*atan2(S1(1),S1(2))+pi/2;

% Analisando os Elos

% Elo 3

V1 = [Z1(1)-S1(1) Z1(2)-S1(2)];
V1_mag = sqrt(V1(1)^2 + V1(2)^2);
t3 = (-1)*atan2(V1(1),V1(2))+pi/2;

% Elo 4

G1 = [W1(1)+V1(1)-U1(1) W1(2)+V1(2)-U1(2)];
G1_mag = sqrt(G1(1)^2 + G1(2)^2);
t1 = (-1)*atan2(G1(1),G1(2))+pi/2;

% Vetores e Angulos Finais

W = sqrt( (x1(1,1)^2) + (x1(2,1)^2) );
Z = sqrt( (x1(3,1)^2) + (x1(4,1)^2) );
U = sqrt( (x2(1,1)^2) + (x2(2,1)^2) );
S = sqrt( (x2(3,1)^2) + (x2(4,1)^2) );
```

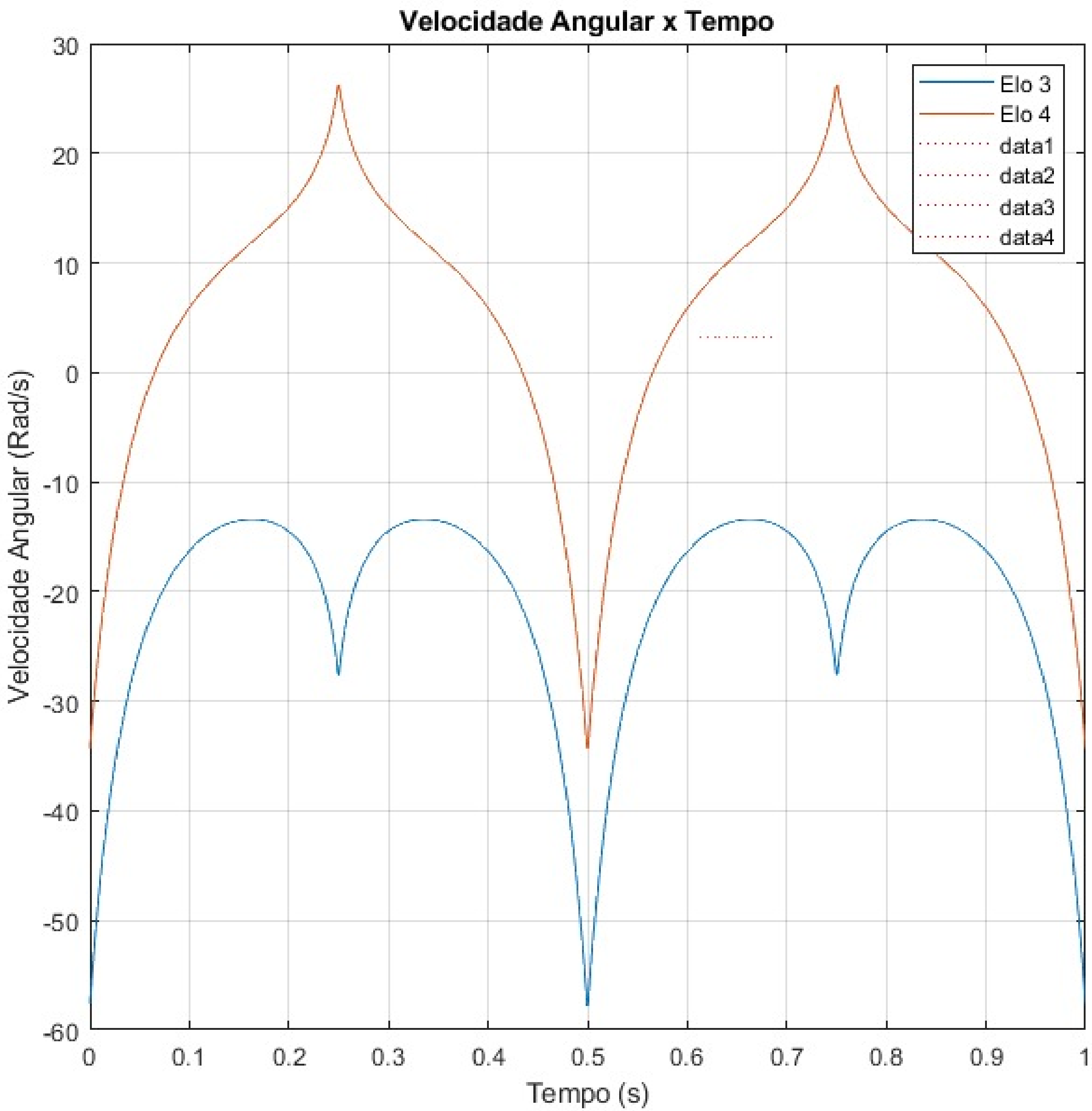
```
W_1x = x1(1,1); W_1y= x1(2,1);
Z_1x = x1(3,1); Z_1y= x1(4,1);
U_1x = x2(1,1); U_1y= x2(2,1);
S_1x = x2(3,1); S_1y= x2(4,1);

vetW = [W_1x W_1y];
vetZ = [Z_1x Z_1y];
vetU = [U_1x U_1y];
vetS = [S_1x S_1y];

Vx = Z_1x - S_1x;
Vy = Z_1y - S_1y;
V = [Vx Vy];
Gx = W_1x + Vx - U_1x;
Gy = W_1y +Vy - U_1y;
G = norm([Gx Gy]);

theta3=atan2(Vy,Vx);
t2i = t - t1;
t2f = t2i + beta(3);

% Condicao Grashof
barras = [W U V G];
s = min(barras); l = max(barras);
pq = W + U + V + G - (s + l);
sl = s + l;

if sl <= pq
    disp("Grashof");
else
    disp("Nao Grashof");
end

% Localizacao dos pivos
O2x = -Z*cos(phi) - W*cos(theta3);
O2y = -Z*sin(phi) - W*sin(theta3);
O4x = -S*cos(psi) - U*cos(sigma);
O4y = -S*sin(psi) - U*sin(sigma);
thetarot = (-1)*atan2((O4x - O2x), (O4y - O2y)) + pi/2;

% Comprimento dos elos
a=W; b=norm(V); c=U; d=G; e=norm(vetZ);

% Numero de passos
N=1000;
t=linspace(0,1,N)';

% Analise do movimento
w2=4*pi;
t2v = [linspace(t2i,t2f,N/4) linspace(t2f,t2i,N/4) ...
       linspace(t2i,t2f,N/4) linspace(t2f,t2i,N/4)]';

% Metodo de Newton-Raphson para determinacao dos angulos t3 e t4
tol=0.001;
```

```
t3=theta3;
t4=acos((S.^2+V.^2-Z.^2)/(2*S*V));
for it2=1:length(t2v)
    t2=t2v(it2); B=tol+1; iconv=0;
    while norm(B)>tol
    iconv=iconv+1;
    A=[-b*sin(t3) c*sin(t4);b*cos(t3) -c*cos(t4)];
    B=[a*cos(t2)+b*cos(t3)-c*cos(t4)-d; a*sin(t2)+b*sin(t3)-c*sin(t4)];
    Dt=-A\B;
    t3=t3+Dt(1); t4=t4+Dt(2);
    end
    t3v(it2,1)=t3;
    t4v(it2,1)=t4;
end

% Declaracao dos vetores velocidade angular e aceleracao angular
w3=[]; w4=[];
a3=[]; a4=[];
% Declaracao dos vetores velocidade linear e aceleracao linear
Va=[]; Vb=[];
Aa=[]; Ab=[];

for i=1:length(t)
    Theta2=t2v(i);
    Theta3=t3v(i);
    Theta4=t4v(i);
    % Usando as equacoes da dinamica
    w3(i)=(a*w2/b)*(sin(Theta4-Theta2)/sin(Theta3-Theta4));
    w4(i)=(a*w2/c)*(sin(Theta2-Theta3)/sin(Theta4-Theta3));

    % Constantes necessarias para o calculo das aceleracoes angulares
    A=c*sin(Theta4);
    B=b*sin(Theta3);
    C=a*0*sin(Theta2)+a*(w2^2)*cos(Theta2)+b*(w3(i)^2)*cos(Theta3)-c*(w4(i)^2)*cos↙
(Theta4);
    D=c*cos(Theta4);
    E=b*cos(Theta3);
    F=a*0*cos(Theta2)-a*(w2^2)*sin(Theta2)-b*(w3(i)^2)*sin(Theta3)+c*(w4(i)^2)*sin↙
(Theta4);

    a3(i)=(C*D-A*F)/(A*E-B*D);
    a4(i)=(C*E-B*F)/(A*E-B*D);

    % Calculando cada componente das velocidades lineares e depois seu modulo
    Va_i=a*w2*(-sin(Theta2));
    Va_j=a*w2*(cos(Theta2));
    Va(i)=((Va_i^2)+(Va_j^2))^(1/2);
    Vb_i=c*w4(i)*(-sin(Theta4));
    Vb_j=c*w4(i)*(cos(Theta4));
    Vb(i)=((Vb_i^2)+(Vb_j^2))^(1/2);

    % Calculando cada componente das aceleracoes lineares e depois seu modulo

    Aa_i=-(a*0*sin(Theta2)+a*(w2^2)*cos(Theta2));
    Aa_j=a*0*cos(Theta2)-a*(w2^2)*sin(Theta2);
```

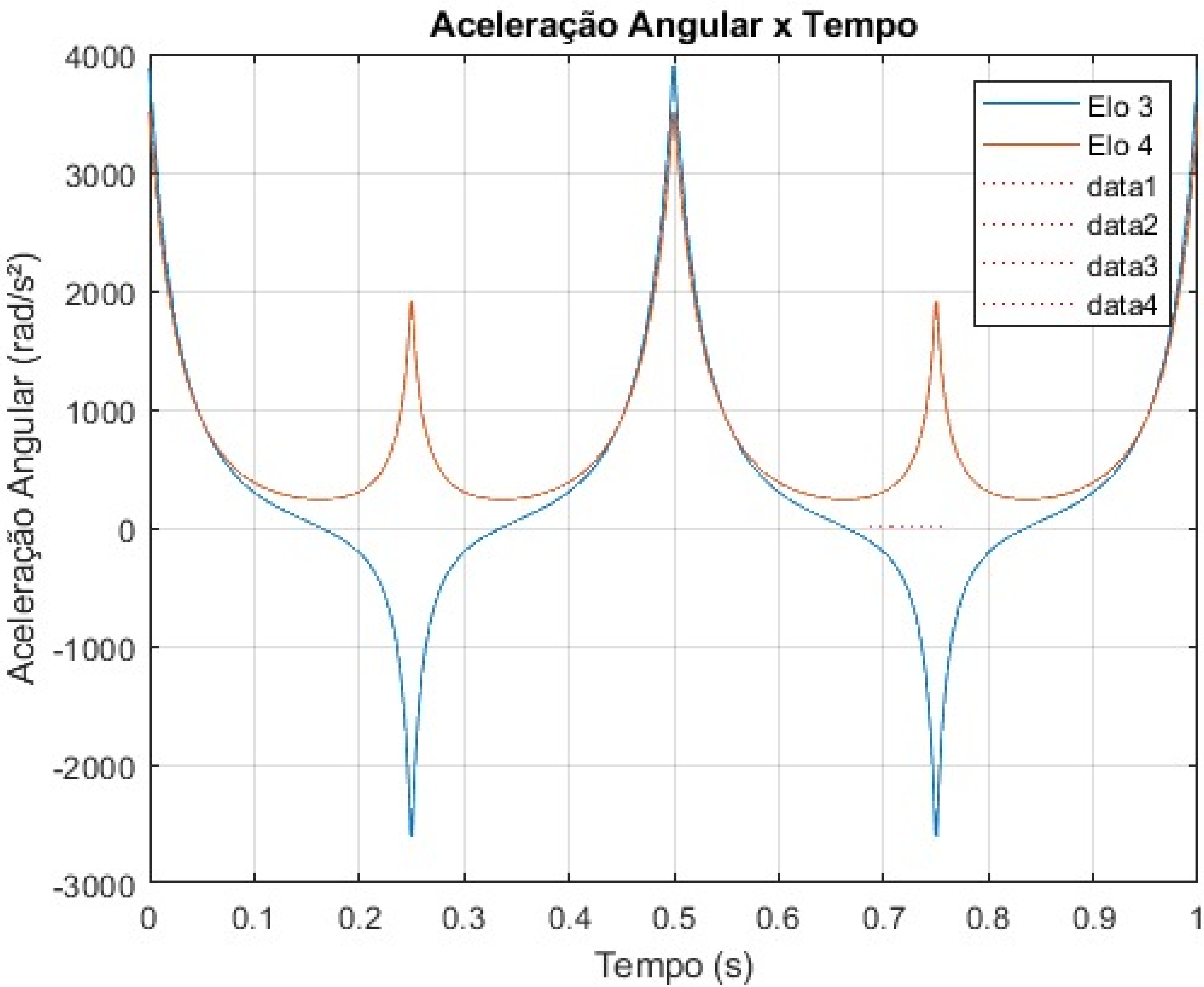
```
    Aa(i)=((Aa_i^2)+(Aa_j^2))^(1/2);
    Ab_i=-(c*a4(i)*sin(Theta4)+c*(w4(i)^2)*cos(Theta4));
    Ab_j=c*a4(i)*cos(Theta4)-c*(w4(i)^2)*sin(Theta4);
    Ab(i)=((Ab_i^2)+(Ab_j^2))^(1/2);
end

% Plotagem dos graficos das velocidades angulares w3 e w4
figure(1)
plot(t,w3);
xlabel('Tempo (s)');
ylabel('Velocidade Angular (Rad/s)');
title('Velocidade Angular x Tempo');
grid on;
hold on;
plot(t,w4);
legend('Elo 3','Elo 4');
axis([0 1 -60 30]);
hold off;

% Plotagem dos graficos das aceleracoes angulares a3 e a4
figure(2)
plot(t,a3);
xlabel('Tempo (s)');
ylabel('Aceleração Angular (rad/s²)');
title('Aceleração Angular x Tempo');
grid on;
hold on
plot(t,a4);
legend('Elo 3','Elo 4');
axis([0 1 -3000 4000]);
hold off;

% Plotagem dos graficos das velocidades lineares Va e Vb
figure(3)
plot(t,Va);
xlabel('Tempo (s)');
ylabel('Velocidade Linear (m/s)')
title('Velocidade Linear x Tempo');
grid on;
hold on;
plot(t,Vb);
legend('Ponto A','Ponto B');
axis([0 1 0 150]);
hold off;

% Plotagem dos graficos das aceleracoes lineares Aa e Ab
figure(4)
plot(t,Aa);
xlabel('Tempo (s)');
ylabel('Aceleração Linear (m/s²)');
title('Aceleração Linear x Tempo');
grid on;
hold on;
plot(t,Ab);
legend('Ponto A','Ponto B');
```

```
axis([0 1 0 12000]);
hold off;

% Evaluate positions of points A,B,C,D,P
AP=Z1;
tAP=S1;
rA=zeros(length(t2v),2);
rB=a*[cos(t2v) sin(t2v)];
rC=rB+b*[cos(t3v) sin(t3v)];
rD=[rA(:,1)+d rA(:,2)];
rP=rB+Z1.*[cos(t3v+(phi-theta3)) sin(t3v+(phi-theta3))];
tol=0.003;
posP2 = find(t2v>t2i+beta(2)-tol & t2v>t2i+beta(2)+tol);

% Show simulation
outvid=0;
figure(1), clf, set(1,'position',[0 0 690 650])
    if outvid==1
        filename='analuiza-pr tica5.gif';
        frame=getframe(gcf); im=frame2im(frame); [A,map]=rgb2ind(im,256);
        imwrite(A,map,filename,'gif','LoopCount',Inf,'DelayTime',1e-3);
    end
    rT=[rA; rB; rC; rD; rP]; mx=max(max(abs(rT)));
    rT=[rA; rB; rC; rD; rP]; mx=max(max(abs(rT))); axis([-mx mx -mx mx]),
    axis equal,
    hAB=line([rA(1,1) rB(1,1)],[rA(1,2) rB(1,2)]); set(hAB,'Color',.7*[1 1↙
1],'LineStyle','-'),
    hBC=line([rB(1,1) rC(1,1)],[rB(1,2) rC(1,2)]); set(hBC,'Color',.7*[1 1↙
1],'LineStyle','-'),
    hCD=line([rC(1,1) rD(1,1)],[rC(1,2) rD(1,2)]); set(hCD,'Color',.7*[1 1↙
1],'LineStyle','-'),
    hDA=line([rD(1,1) rA(1,1)],[rD(1,2) rA(1,2)]); set(hDA,'Color',.7*[1 1↙
1],'LineStyle','-'),
    hBP1=line([rB(1,1) rP(1,1)],[rB(1,2) rP(1,2)]); set(hBP1,'Color',.7*[1 1↙
1],'LineStyle','-'),
    hBP2=line([rB((length(rB)/4),1) rP((length(rP)/4),1)],[rB((length(rB)/4),2) rP↙
((length(rP)/4),2)]); set(hBP2,'Color',.7*[1 1 1],'LineStyle','-'),
    hBP3=line([rB(posP2(1,1),1) rP(posP2(1,1),1)],[rB(posP2(1,1),2) rP(posP2(1,1),↙
2)]); set(hBP3,'Color',.7*[1 1 1],'LineStyle','-'),
    hBP1=line([rB(1,1) rP(1,1)],[rB(1,2) rP(1,2)]); set(hBP1,'Color',.7*[1 1↙
1],'LineStyle','-'),

    text(rA(1,1)-mx/100,rA(1,2)-mx/20,'A')
    text(rB(1,1)-mx/100,rB(1,2)-mx/20,'B')
    text(rC(1,1)-mx/100,rC(1,2)-mx/20,'C')
    text(rD(1,1)-mx/100,rD(1,2)-mx/20,'D')

    if outvid==1 dts=5; else dts=1; end
    for n=2:dts:length(t)
        set(hAB,'xdata',[rA(n,1) rB(n,1)],'ydata',[rA(n,2) rB(n,2)]);
        set(hBC,'xdata',[rB(n,1) rC(n,1)],'ydata',[rB(n,2) rC(n,2)]);
        set(hCD,'xdata',[rC(n,1) rD(n,1)],'ydata',[rC(n,2) rD(n,2)]);
        set(hBP1,'xdata',[rB(n,1) rP(n,1)],'ydata',[rB(n,2) rP(n,2)]);
        hP=line([rP(n-1,1) rP(n,1)],[rP(n-1,2) rP(n,2)]); set↙
(hP,'Color','r','LineStyle',':');
```

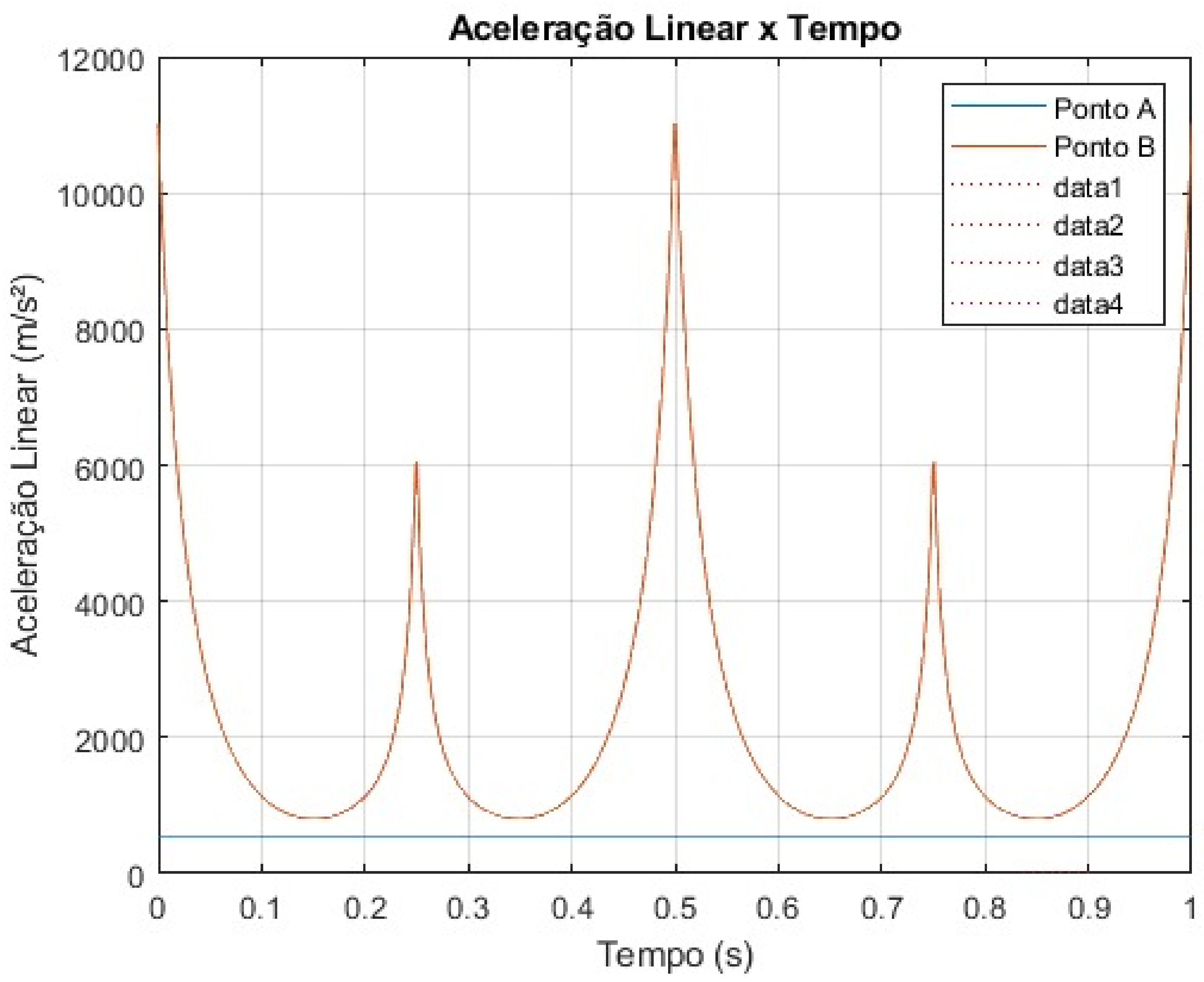
```
    if outvid==1
        frame=getframe(gcf); im=frame2im(frame); [A,map]=rgb2ind(im,256);
        imwrite(A,map,filename,'gif','WriteMode','append','DelayTime',1e-3);
    else pause(1e-12);
    end
end
```

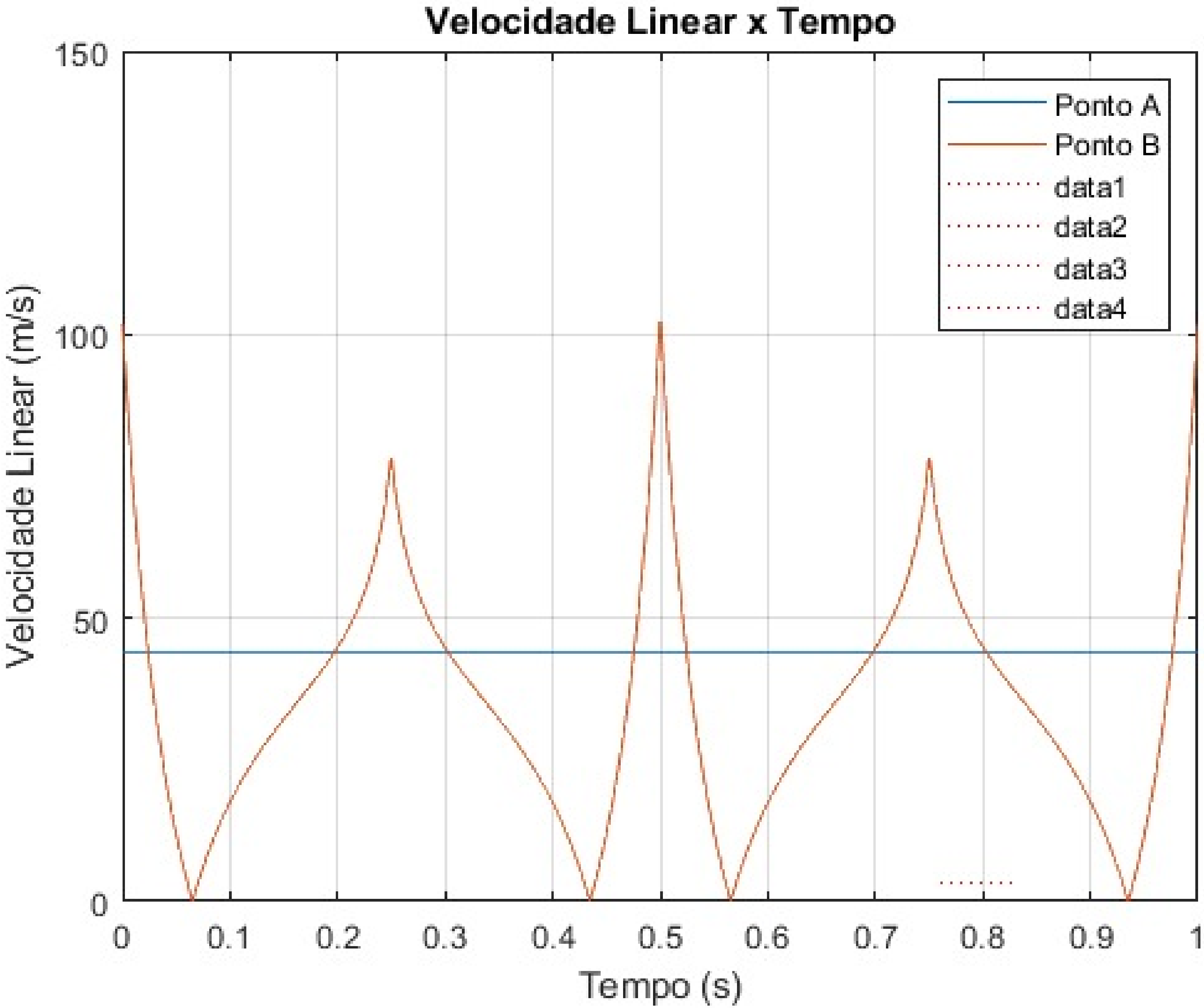


Velocidade Angular x Tempo
Elo 3
Elo 4
data1
data2
data3
data4
30
20
10
0
-10
-20
-30
-40
-50
-60
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Velocidade Angular (Rad/s)
Tempo (s)



Aceleração Angular x Tempo
Aceleração Angular (rad/s²)
Tempo (s)
Elo 3
Elo 4
data1
data2
data3
data4
4000
3000
2000
1000
0
-1000
-2000
-3000
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Aceleração Linear x Tempo
Aceleração Linear (m/s²)
Tempo (s)
Ponto A
Ponto B
data1
data2
data3
data4
12000
10000
8000
6000
4000
2000
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1



Video da Similação:
https://drive.google.com/file/d/1SEHwajbjRoqBJeb7FgK04S9DWjFZHXgL/view?usp=sharing